Fig. 1

Fig. 2

260

Address Range 502	Home 504	Mode 506
A	Client 3	PTP
B	Client 3	BC
C	Client 1	PTP
D	Client 4	PTP
E	Client 3	BC
F	Client 2	BC
G	Client 5	PTP
⋮	⋮	⋮

510A — 510B — 510C — 510D — 510E — 510F — 510G

Fig. 3

220A

Address 602	Client 1 604	Client 2 606	Client 3 608	Client 4 610	Client 5 612	Owner 614
Aa	—	—	M	—	—	Client 3
Ab	—	—	M	—	—	Client 3
Ac	—	—	M	—	—	Client 3
Ad	O	—	S	S	—	Client 1
Ae	S	—	S	S	—	None
Af	S	—	O	—	—	Client 3
Ag	—	—	—	M	—	Client 4
. . .	. . .	. . .	. . .	. . .	. . .	. . .

620A
620B
620C
620D
620E
620F
620G

Fig. 4

220B

Address 602	Client 1 604	Client 2 606	Client 3 608	Client 4 610	Client 5 612	Owner 614
Aa	—	—	W	—	—	Client 3
Ab	—	—	W	—	—	Client 3
Ac	—	—	W	—	—	Client 3
Ad	R	—	R	R	—	Client 1
Ae	R	—	R	R	—	None
Af	R	—	R	—	—	Client 3
Ag	—	—	—	W	—	Client 4
· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·	· · · ·

620A
620B
620C
620D
620E
620F
620G

Fig. 4A

```
            ┌─────────────────────┐
            │ Configure Address   │
            │     Network         │
            │      300            │
            └──────────┬──────────┘
                       │
                       ▼
                    ╱──────╲
                  ╱  Address  ╲
                ╱ Network receives ╲     No
               ╲  Client Request   ╱─────────
                ╲      302       ╱
                  ╲            ╱
                    ╲────────╱
                       │ Yes
                       ▼
            ┌─────────────────────┐
            │  Determine mode     │
            │      304            │
            └──────────┬──────────┘
                       │
                       ▼
  ┌──────────────┐  ╱──────╲  No  ┌──────────────────┐
  │ Broadcast    │ ╱ Mode =  ╲────▶│ Convey Request   │
  │ Request  ◀───┤╲  BC?    ╱ Yes │ Point-to-Point   │
  │   308        │ ╲  306  ╱       │     310          │
  └──────────────┘  ╲────╱        └──────────────────┘
                      Yes
```

Fig. 5

```
          ┌─────────────────┐
          │ Initialize Address │
          │     Network     │
          │       400       │
          └─────────────────┘
                   │
                   ▼
              ╱ Condition ╲        No
            ◄  Detected   ►─────────►
              ╲    402    ╱
                   │
                  Yes
                   │
                   ▼
          ┌─────────────────┐
          │ Suspend Clients │
          │       404       │
          └─────────────────┘
                   │
                   ▼
          ┌─────────────────┐
          │    Complete     │
          │   outstanding   │
          │   transactions  │
          │       406       │
          └─────────────────┘
                   │
                   ▼
          ┌─────────────────┐
          │   Change Mode   │
          │       408       │
          └─────────────────┘
```

Fig. 6

Packet Type	Full Name	Address Space		Description
		Cacheable	I/O	
RTS	ReadToShare	Y		Requests read-only copy of cache line
RTO	ReadToOwn	Y		Requests writable copy of cache line
RTWB	ReadToWriteBack	Y		Requests to receive writable copy of cache line and send cache line to memory
RS	ReadStream	Y		Request read-once copy of cache line
WS	WriteStream	Y		Request to write entire cache line and send to memory
WB	WriteBack	Y		Request to send cache line from owning device to memory, device does not keep copy
WBS	WriteBackShared	Y		Request to send cache line from owning device to memory, device keeps read-only copy
RIO	ReadIO		Y	Request to read IO locations
WIO	WriteIO		Y	Request to write IO locations
INT	Interrupt			Sends an interrupt, target is specified by address

Fig. 7

Transaction Type	Initiator Receives	Initiator Sends
RTS	DATA	
RTO	DATA	
RTWB	DATA & PRN	DATA
RS	DATA	
WS	ACK & PRN	DATA
WB	PRN	DATA or NACK
WBS	PRN	DATA or NACK

Fig. 8

Transaction Type	Initiator Receives	Initiator Sends
RIO	DATA	
WIO	PRN	DATA
INT	PRN or NACK	DATA or Nothing

Fig. 9

Access Rights Symbol	Access Rights Name	Description	Data Present?
W	Write	Read and Write	Yes
A	All-Write	Write-only, must write entire cache line	Yes (or ACK)
R	Read	Read-only	Yes
T	Transient-Read	Read-only, read can be reordered	Yes
I	Invalid	No access rights	Yes or No

Fig. 10A

Ownership Status Symbol	Ownership Status Name	Description	Data Present?
O	Owner	Owns cache line	Yes or No
N	Not Owner	Does not own cache line	Yes or No

Fig. 10B

Access Right	Ownership Status
W	O
R	O
I	O
W	N
A	N
R	N
T	N
I	N

Fig. 10C

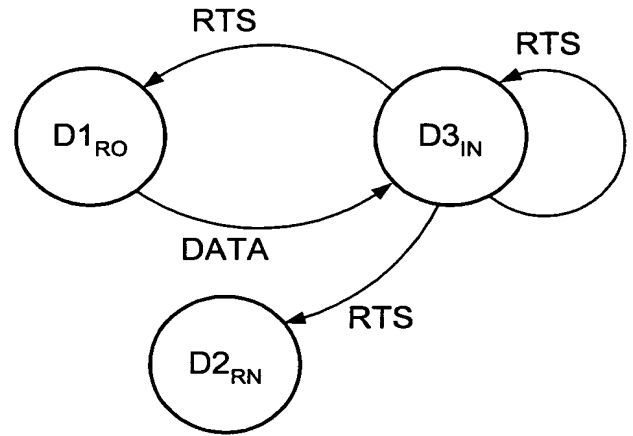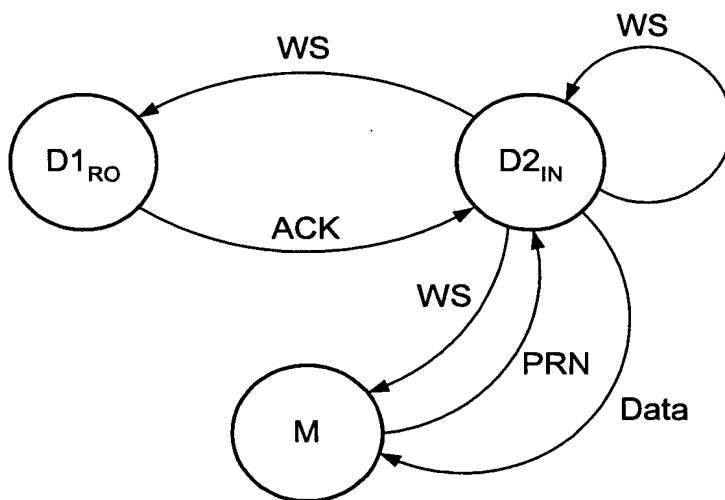| Transaction Type	New Owner
RTS	Previous Owner
RTO	Initiator
RTWB	Memory
RS	Previous Owner
WS	Memory
WB	Memory (or Previous Owner)
WBS	Memory (or Previous Owner)

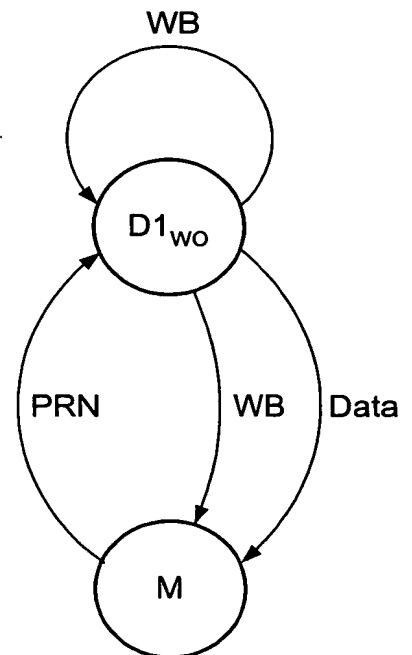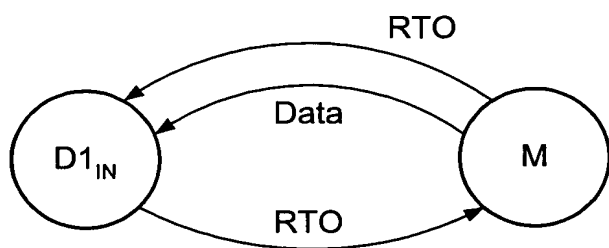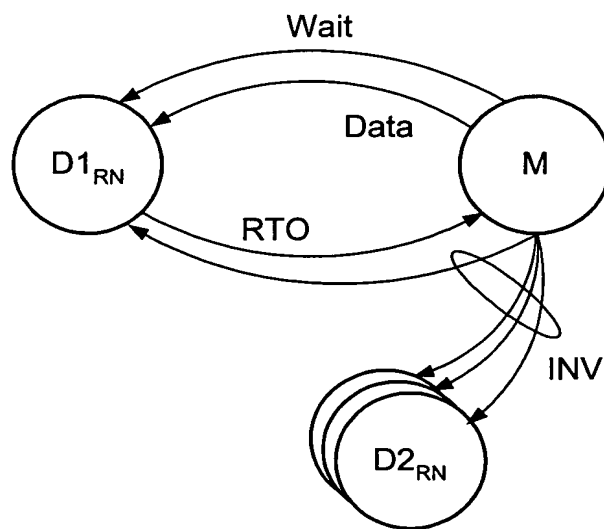*Fig. 11*
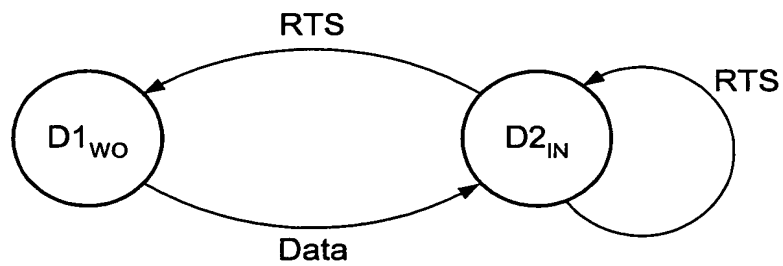
Fig. 12A
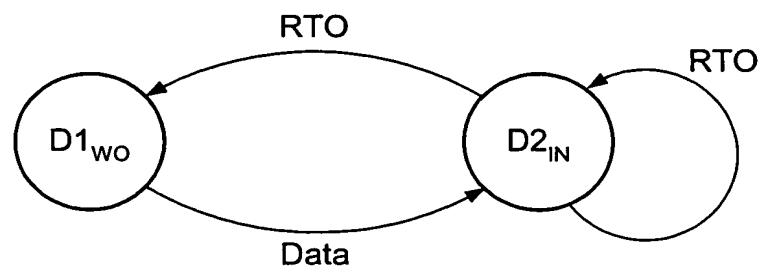
Fig. 12B
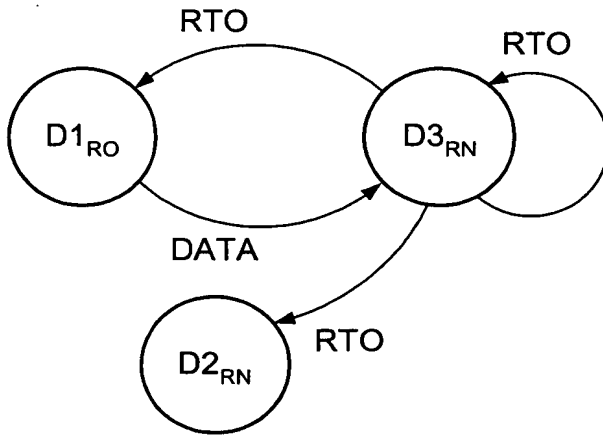
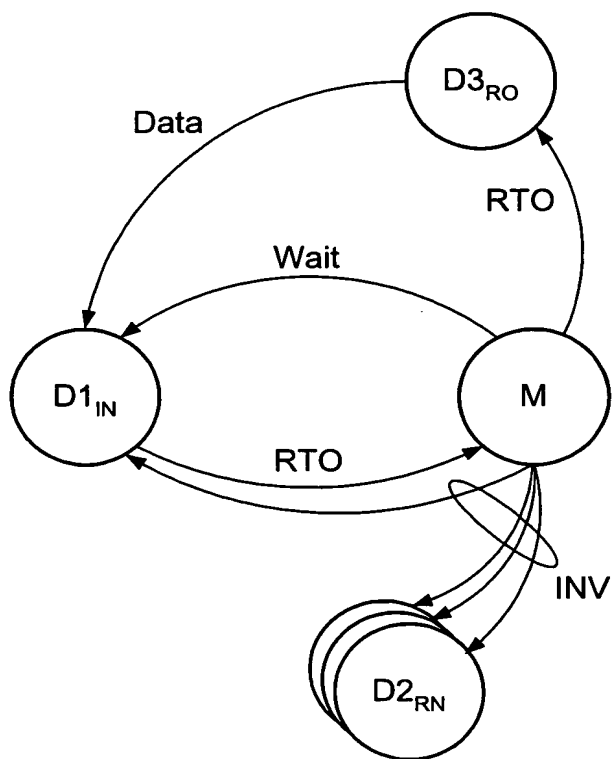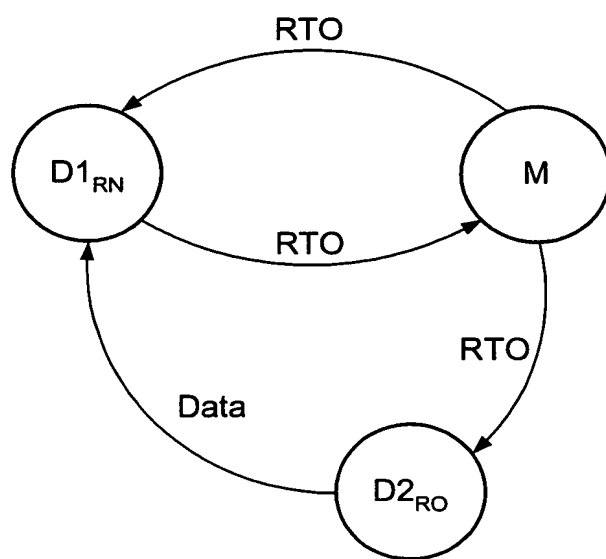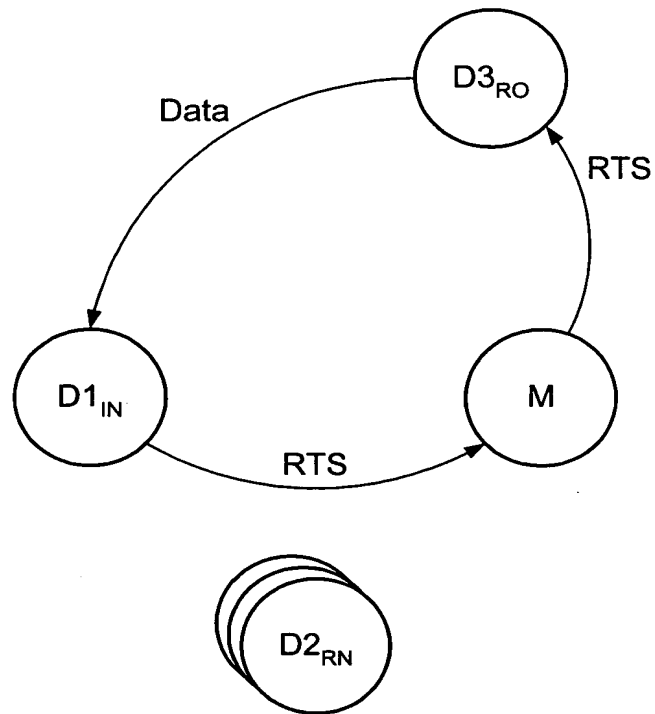*Fig. 12C*

Fig. 12D

Fig. 12E

Fig. 12F

Fig. 13A

Fig. 13B

Fig. 13C

Fig. 13D

Fig. 13E
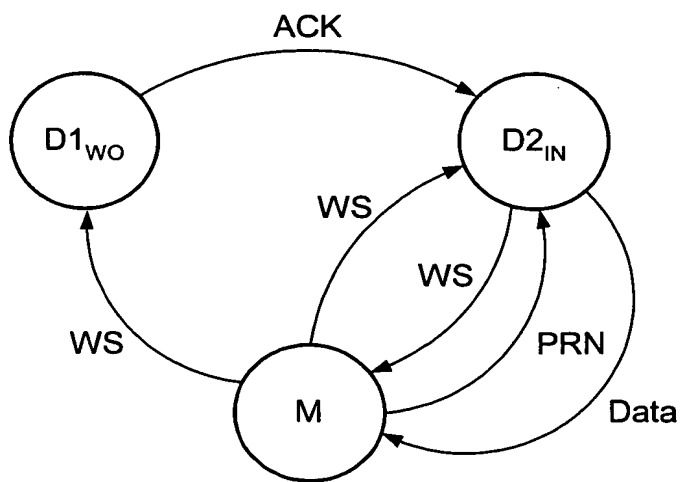
Fig. 13F
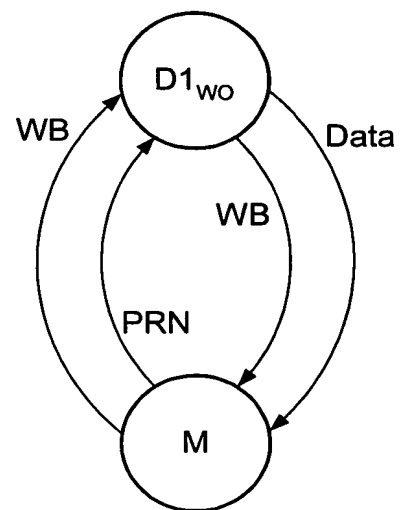
Fig. 13G
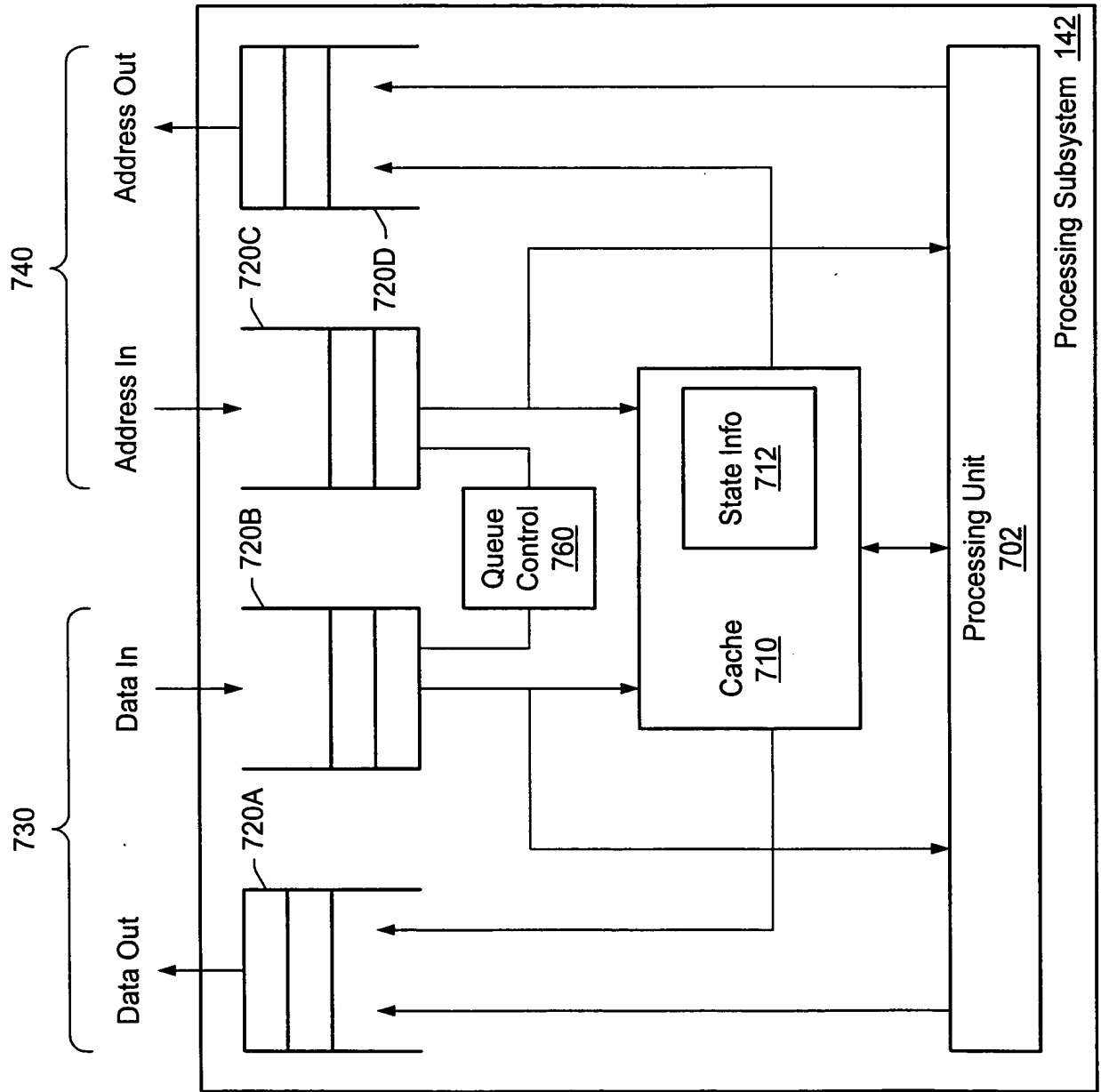
Fig. 14
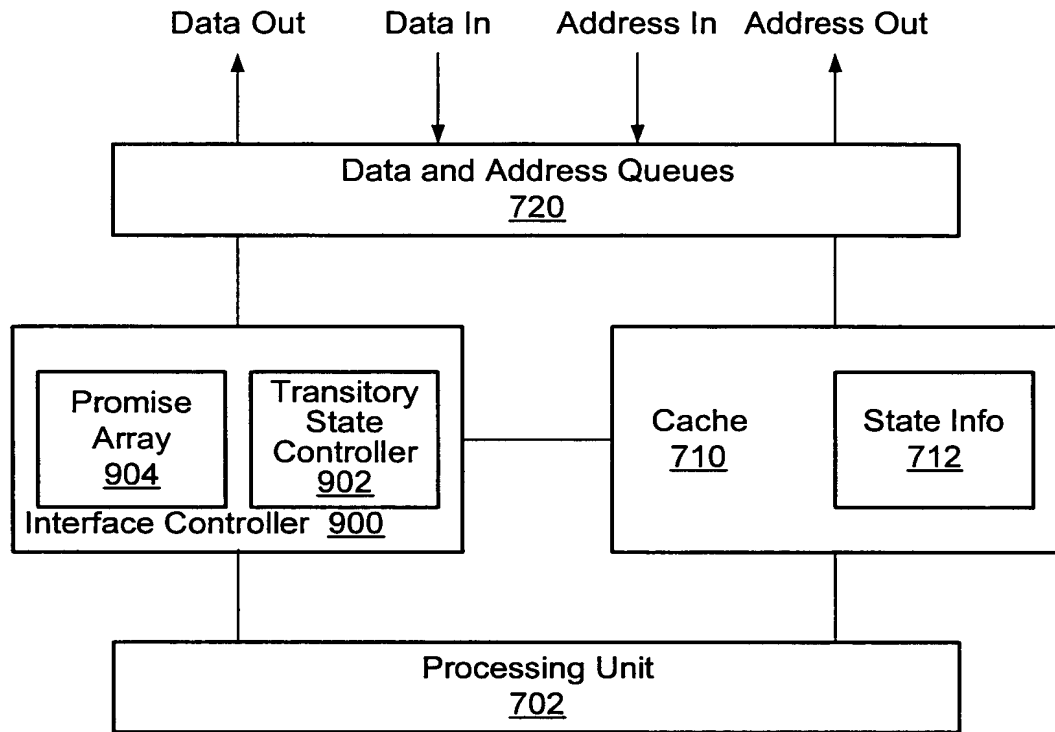
Fig. 15

State	Description
WOfh	Waiting for WAIT and ACK for local WS, has write access and ownership
WO	Stable state with write access and ownership
ROfh	Waiting for WAIT and ACK for local WS, has read access and ownership
ROe	Waiting for DATA for local RTO, has read access and ownership
ROde	Waiting for INV and DATA for local RTO, has read access and ownership
ROd	Waiting for INV for local RTO, has read access and ownership
ROce	Waiting for WAIT and DATA for local RTO, has read access and ownership
RO	Stable state with read access and ownership
IOe	Waiting for DATA for local RTO, has no access rights, has ownership
IOde	Waiting for INV and DATA for local RTO, has no access rights, has ownership
IOd	Waiting for INV for local RTO, has no access rights, has ownership
WNj	Able to send DATA/NACK for local WB/WBS, has write access and no ownership
WNi	Able to send DATA for local RTWB, write access, no ownership
WNh	Waiting for ACK for local WS, has write access, no ownership
WNe	Waiting for WAIT for local RTO, has write access, no ownership
WN	Stable state with write access, no ownership (caused by foreign transaction that took ownership, but for which no copyback has yet been performed)
ANi	Able to send DATA for local WS after performing write to entire cache line, no ownership
RNj	Able to send DATA/NACK for local WB/WBS, has read access and no ownership
RNh	Waiting for ACK for local WS, has read access, no ownership
RNg	Waiting for INV for local WS, has read access, no ownership
RNe	Waiting for DATA for local RTO, has read access, no ownership
RNde	Waiting for INV and DATA for local RTO, has read access, no ownership
RNd	Waiting for INV for local RTO, has read access, no ownership
RNce	Waiting for WAIT and DATA for local RTO, has read access, no ownership
RNcd	Waiting for WAIT and INV for local RTO, has read access, no ownership
RN	Stable state with read access, no ownership
TN	Performing read(s), which may be reordered, for local RTS or RS, no ownership
INk	Waiting for RTO, RTWB, WS, WB or WBS after receiving an ERR or ERRL. No access rights, no ownership
INj	Able to send DATA/NACK for local WB/WBS, has no access and no ownership
INh	Waiting for ACK for local WS or for DATA for local RTWB, has no access rights, no ownership
INg	Waiting for INV for local WS or for INV for local RTWB, has no access rights, no ownership
INe	Waiting for DATA for local RTO, has no access rights, no ownership
INde	Waiting for INV and DATA for local RTO, has no access rights, no ownership
INd	Waiting for INV for local RTO, has no access rights, no ownership
INce	Waiting for WAIT and DATA for local RTO, has no access rights, no ownership
INcd	Waiting for WAIT and INV for local RTO, has no access rights, no ownership
INa	Waiting for DATA for local RTS, DATA may grant read access, has no access rights, no ownership
IN	Stable state with no access rights, no ownership

Fig. 15A

Action Code	Meaning	Comments
/a	Commit to send an ACK packet as a copyback by appending an entry for the received foreign packet in a copyback list. Set copy tag to I.	ACK packet may be sent from any state that allows copyback packets to be sent. It must be sent within a finite time of first entering such a state, regardless of what other packets have been received.
c/	Commit to send DATA and/or ACK packets for all outstanding copybacks for this cache line. Next, set copy tag to W. Then, perform state transition based on current state & local packet being received	If sending copybacks changes the state from a state X to a state Y, the local packet being received will be received in state Y (and as a result, the entry for state Y in the table must be consulted to determine the state transition caused by receiving the local packet).
/d	Commit to send DATA packet for local RTWB, WS, WB, or WBS transaction. DATA packet is sent in response to receiving a PRN packet for this transaction.	DATA packet may not be sent until a PRN packet is received. It must be sent within finite time of receiving the PRN packet & having entered a state that permits the packet to be sent, regardless of what other packets have been received.
e/	Clear outstanding copyback commitments for this line by removing them from the copyback list. Do not send DATA or ACK packets for entries that were on the copyback list. Next, perform state transition based on current state & local packet being received.	The action code is used in response to receiving an ERR or ERRL packet. If an ERR packet was received in place of a PR or PRACK packet, or if an ERRL packet was received in place of a DATAP packet, a DATA packet may be sent to the error device.
/i	Commit to send a DATA packet as a copyback by appending an entry for the received foreign packet in the copyback list. Set copy tag to I.	DATA packet may be sent from any state that allows copyback packets to be sent. It must be sent within finite time of first entering such a state, regardless of what other packets have been received.
/j	Set write tag to I.	
/n	Commit to send NACK packet for local WB or WSB transaction & set write tag to W. NACK packet is sent in response to receiving a PRN packet for this transaction.	NACK packet may be sent at any time after receiving the PRN packet. It must be sent within finite time of receiving the PRN packet, regardless of what other packets have been received.
/r	Commit to send DATA packet as a copyback by appending an entry for the received foreign packet in the copyback list. If copy tag is W, set copy tag to R.	DATA packet may be sent from any state that allows copyback packets to be sent. It must be sent within finite time of first entering such a state, regardless of what other packets have been received.
/s	Set write tag to R.	
/w	Commit to send DATA packet as a copyback by appending an entry for the received foreign packet in the copyback list.	DATA packet may be sent from any state that allows copyback packets to be sent. It must be sent within finite time of first entering such a state, regardless of what other packets have been received.
/y	If copy tag is R, set copy to I.	Used to record invalidating transactions while a copyback for a foreign memory remap is pending.
/z	If write tag is R, set write to I.	Used to record invalidating transactions while a local WBS transaction is pending.

Fig. 15B

Fig. 15C

		Old State →	WOfh	WO	ROfh	ROe	ROde	ROd	ROce	RO	IOe	IOde	IOd	WNi	WNh	WNe	WNj	WN	ANi
		Silent Downgrade		RO														RN	
Send DATA or NACK		Write tag I												IN			IN		IN
		Write tag R															RN	RN	
		Write tag W															WN		
		ERRL for RTO/ RTWB (2)									e/IN								
		ERRL for RTS/RS																	
		ERR (2)													WN		WN		
		WB/WBS		WNi/d						RNi/d							WNj/n		
		ACK for WS (h)																	
		INV for WS (g)		WOfh						ROfh							WNh		
		WAIT for WS (f)	c/ANi/d	c/INg	c/ANi/d					c/INg									
		WS (f+g)		c/ANi/d						c/ANi/d							WNh		
		DATA for RTWB (h)																	
		RTWB/INV for RTWB (g)																	
		DATA for RTO (e)				WO	ROd				WO	IOd							
		INV for RTO (d)					ROe	WO		ROce	IOe	WO							
		WAIT for RTO (c)						c/WO	c/ROd								WO		
		RTO (c+d)						c/WO											
Receive		DATA for RS																	
		DATA for RTS																	
Send on Address Network		WBS		/s						/s							/s		
		WS/WB		/i						/i							/i		
		RTWB																	
		RTO																	
		RS																	
		RTS (a)																	

	INa	Ij	Ij	Is	c/TN	c/IOe	c/IOde	INce	c/INcd	INh	INg	INh	INh	INg	c/INg	INj/n	INk	INk		c/INk	IN	IN	
RNh															ANi/d	c/ANi/d		RN					
RNg															ANi/d								
RNe							RNe																
RNde							$\underline{WN}$ / RNd																
RNd					c/ROe	$\underline{WO}$		c/WNc															
RNce					ROd		WNc																
RNcd					c/ROec/ROde	RNce c/RNcd																	
RNj		Ij	Is	— / —									RNh		RNh	c/RNg	RN	$\underline{RN}$		$\underline{RN}$ $\underline{RN}$			
RN										WNi/d	WNi/d	ANi/d			c/ANi/d	RNj/n		$\underline{RN}$ $\underline{RN}$		$\underline{RN}$	$\underline{IN}$	$\underline{IN}$	
IN					$\underline{c/RN}$								ANi/d				IN	IN	$\underline{IN}$				
INh							c/IOe											e/IN					
INg							IOd																
INe							c/IOe	INe															
INde						$\underline{WO}$		$\underline{WN}$ / INd															
INd						$\underline{IN}$		c/WNc															
INce								WNc															
INcd																							
INk									$\underline{IN}$	$\underline{IN}$		$\underline{IN}$					$\underline{IN}$		$\underline{IN}$				
INj			Ij	Is												$\underline{IN}$							
INa	INa	Ij	Ij	Is	c/TN	c/IOe	c/IOde	INce	c/INcd	INh	INg	INh	INh	INg	c/INg	INj/n	INk	INk	—	c/INk			
IN																							

Fig. 15C (Continued)

Old State	Receive							Send DATA or ACK copyback packet		
	RS	RTS	RTO/ RTWB	WS	INV	MRM	MDM	Copy tag W	Copy tag R	Copy tag I
WOfh	/w	/r	WNh/i	WNh/a					ROfh	
WO	/w	/r	WN/i	WN/a	/z	WN/r	WN/i		RO	
ROfh	/w	/r	RNh/i	RNh/a						
ROe	/w	/r	RNe/i	RNe/a	IOe	RNe/r	RNe/i			
ROde	/w	/r	RNde/i	RNde/a	IOde					
ROd	/w	/r	RNd/i	RNd/a						
ROce	/w	/r	RNce/i	RNce/a						
RO	/w	/r	RN/i	RN/a	/z	RN/r	RN/i			
IOe	/w	/r	INe/i	INe/a		INe/r	INe/i			
IOde	/w	/r	INde/i	INde/a						
IOd	/w	/r	INd/i	INd/a						
WNi										
WNh									RNh	INh
WNc										
WNj		/yz		/yz	/z		/yz		RNj	INj
WN		/y		/y	/z		/y		RN	IN
ANi										
RNh					INh					INh
RNg										
RNe		/y		/y	INe		/y			
RNde					INde					
RNd										
RNce					INce					INce
RNcd										
RNj			INj	INj	INj		INj			INj
RN			IN	IN	IN		IN			IN
TN										
INh										
INg										
INe		/y		/y			/y			
INde										
INd										
INce										
INcd										
INk										
INj										
INa			IN	IN	IN		IN			
IN										

Fig. 15D

Fig. 16

Fig. 17

Fig. 18

Fig. 19

100

140A

Address Network
150A

| Processing Subsystem **142AA** | Processing Subsystem **142BA** | Memory Subsystem **144AA** | Memory Subsystem **144BA** | I/O Subsystem **146A** | Interface **148A** |

Data Network
152A

140B

Address Network
150B

| Processing Subsystem **142AB** | Processing Subsystem **142BB** | Memory Subsystem **144AB** | Memory Subsystem **144BB** | I/O Subsystem **146B** | Interface **148B** |

Data Network
152B

140C

Address Network
150C

| Processing Subsystem **142AC** | Processing Subsystem **142BC** | Memory Subsystem **144AC** | Memory Subsystem **144BC** | I/O Subsystem **146C** | Interface **148C** |

Data Network
152C

Inter-Node Network **154**

Fig. 20

gTag	Description
gM	The maximum access right within the node is Write Access
gS	The maximum access right within the node is Read Access. No client device in the node can have Write Access.
gI	The maximum access right within the node is Invalid Access. No client device in the node can have Read or Write Access.

Fig. 21

Packet Type	Full Name	Address Space		Description
		Cacheable	I/O	
PRTS	Proxy RTS	Y		Request from an interface in a gS or gi node in response to a RTS request from another node
PRTSM	Proxy RTS Modified	Y		Request from an interface in a gM node in response to a RTS request from another node
PRTOM	Proxy RTO Modified	Y		Request from an interface in a gM node in response to a RTO request from another node
PRTO	Proxy ReadToOwn	Y		Request from an interface in response to a RTO request from another node
PU	Proxy Upgrade	Y		Request from an interface asking memory to supply data for an outstanding RTO
PDU	ProxyDataUpgrade	Y		Request from an interface asking memory to update gTag to gM; interface supplies data for an outstanding RTO
PRSM	Proxy ReadStream Modified	Y		Request from an interface in a gM node in response to RS request from another node
PIM	ProxyInvalidate Modified	Y		Request from an interface in a gM node to invalidate data in caches and memory
PI	ProxyInvalidate	Y		Request from an interface in a gS or gl node to invalidate data in caches and memory
PMR	ProxyMemoryRead	Y		Request from an interface to memory to read coherency state(s) and data or meta-data
PMW	ProxyMemoryWrite	Y		Request from an interface to memory to write coherency state(s) and data or meta-data

Fig. 22

Packet Type	Full Name	Description
DATAM	Data-Meta	Data packet containing data and coherence state information
DATAN	Data-NoPull	Data packet sent in response to PRTSM indicating no PRN will be coming
REP	Report	Report from memory to an interface indicating a transaction to be handled by the interface

Fig. 23

Subtransaction	Preexisting Global Access State	New Global Access State
PRTSM	Modified	Shared
PRTOM	Modified	Invalid
PRTO	Shared, Invalid	Invalid
PU	Shared	Modified
PDU	Shared, Invalid, Modified	Modified
PRSM	Modified	Unchanged
PIM	Modified	Invalid
PI	Shared, Invalid	Invalid
PMR	Shared, Invalid, Modified	Unchanged
PMW	Shared, Invalid, Modified	Equal to new global access state specified in DATAM packet

Fig. 24
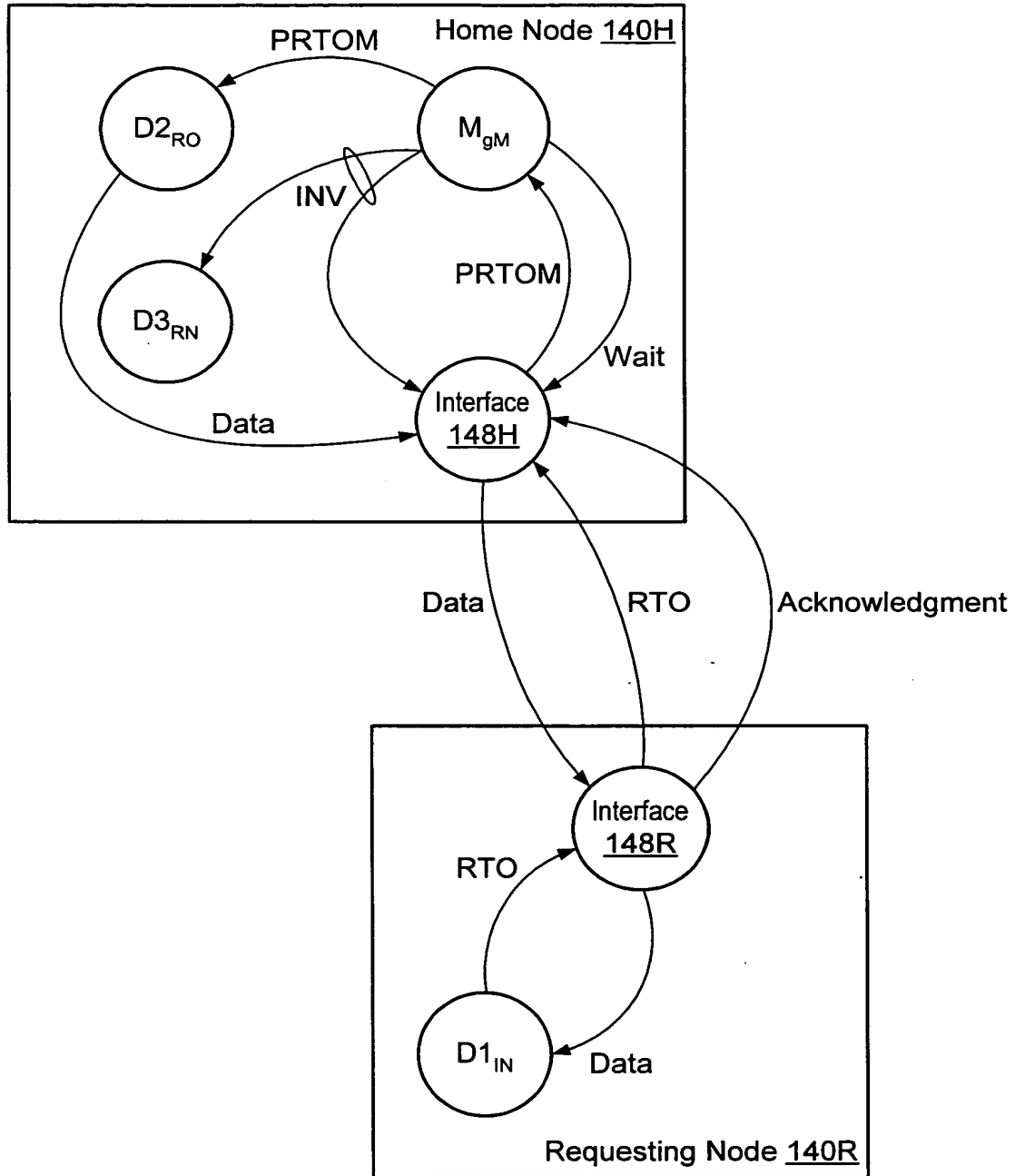
Fig. 25
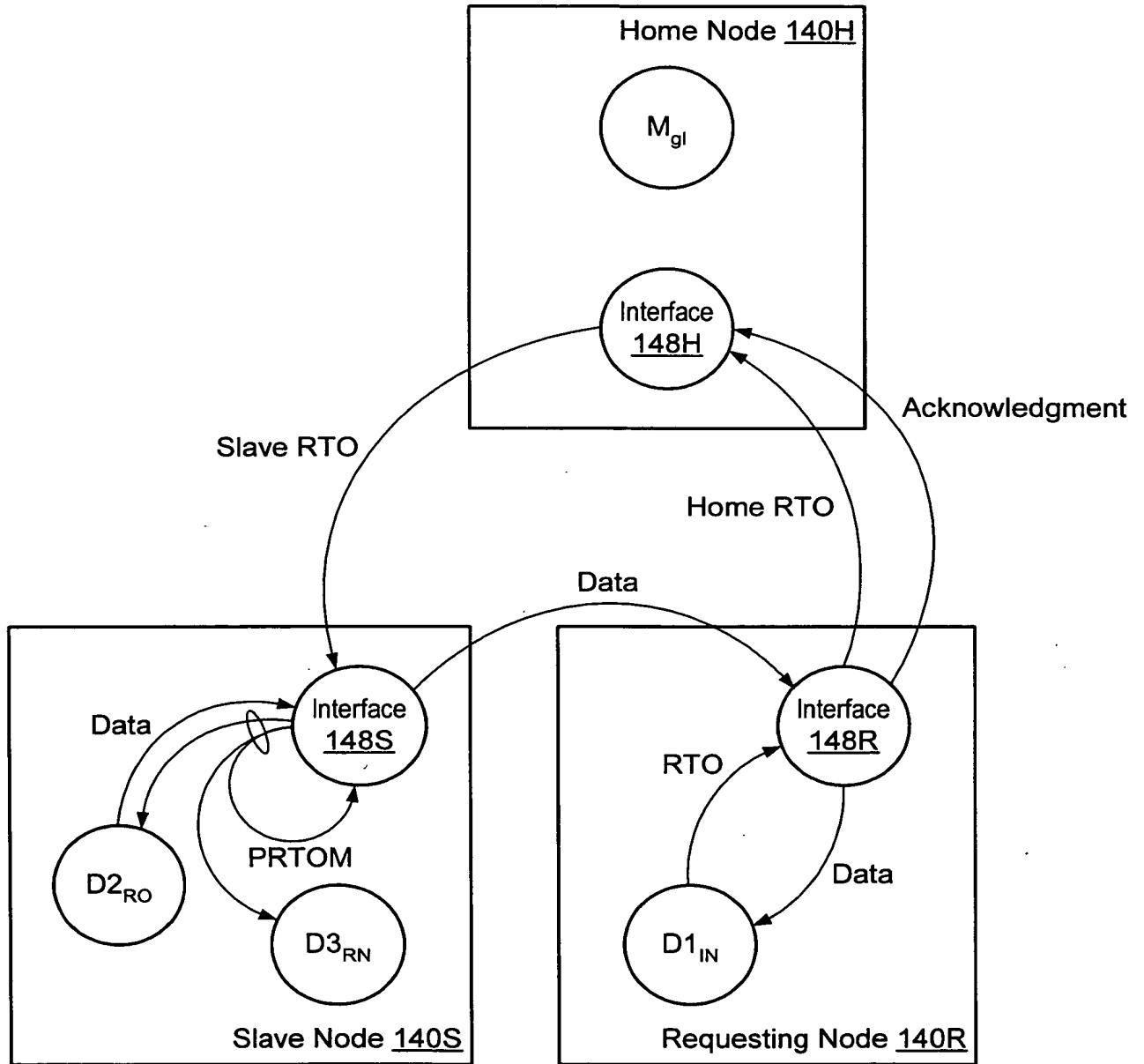
Fig. 26
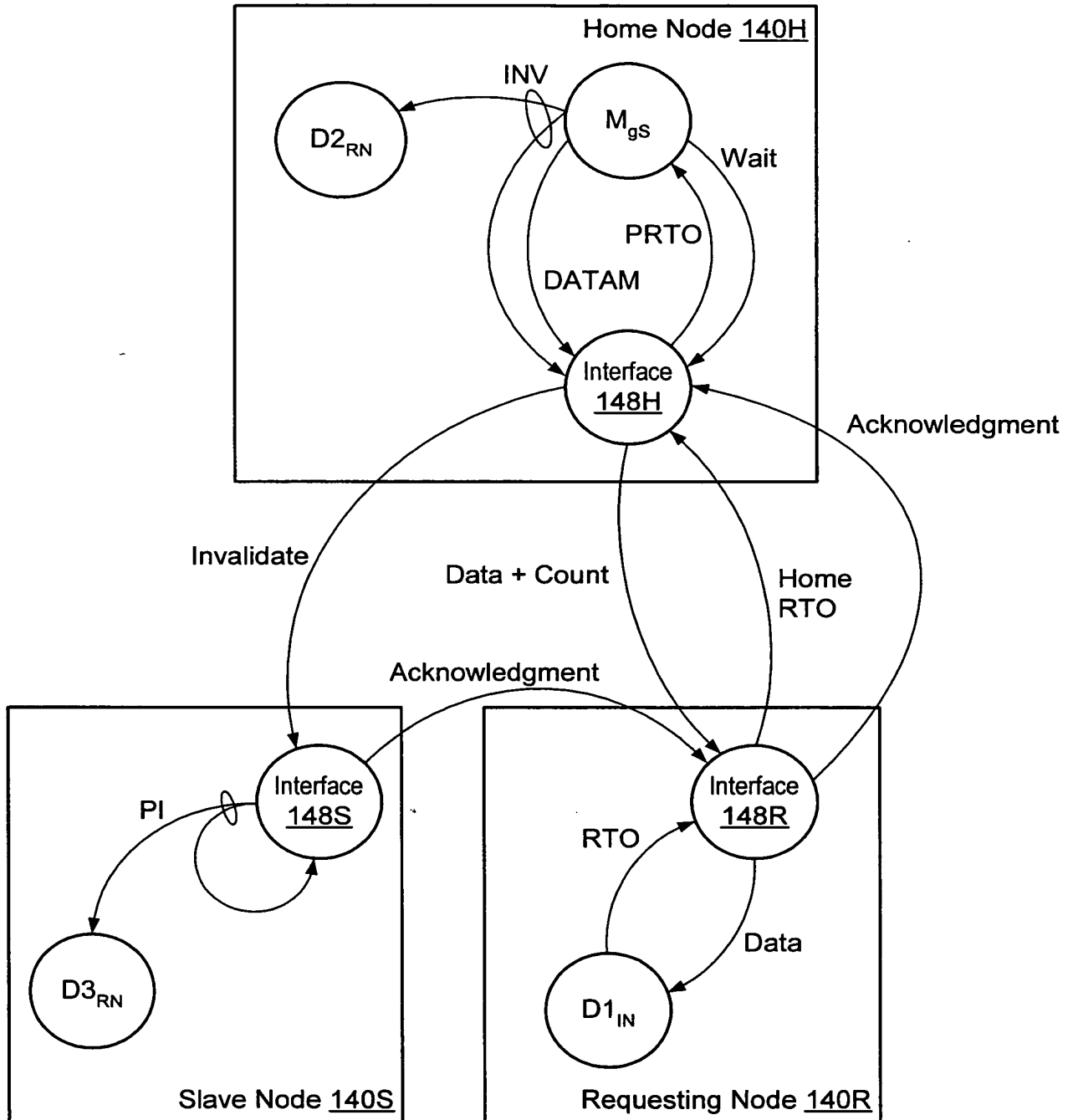
Fig. 27
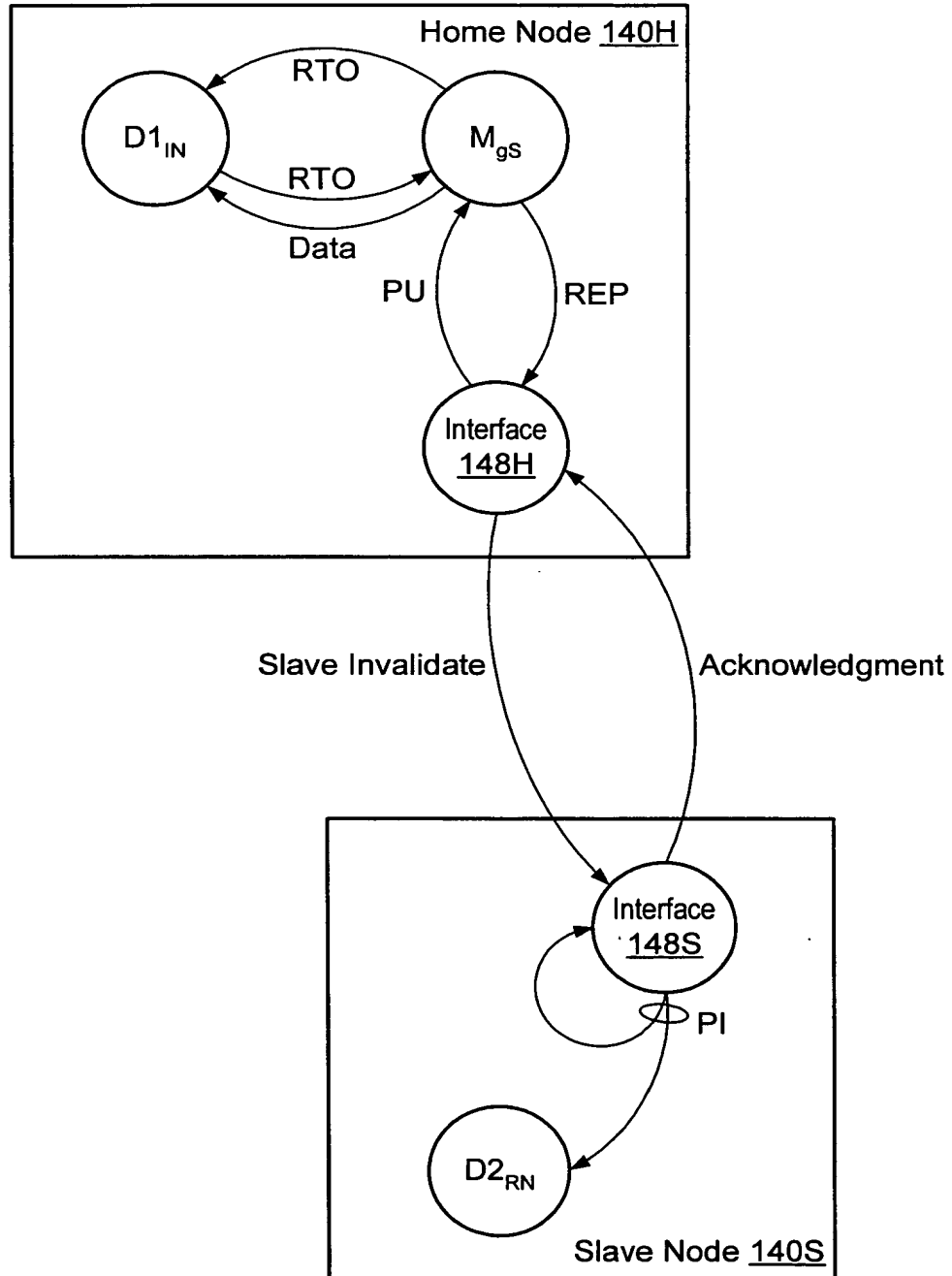
Fig. 28

Fig. 29

Fig. 30

Fig. 31

Fig. 32

Home Node 140H

PRTO

$D2_{RO}$

PRTOM

PRTOM

M_{gM}

PRTO

PRTO

Data

PRTOM

Interface
148H

DATAM

Acknowledgment

$D3_{RN}$

Data

Home RTO

Interface
148R

RTO

Data

$D1_{IN}$

Requesting Node 140R
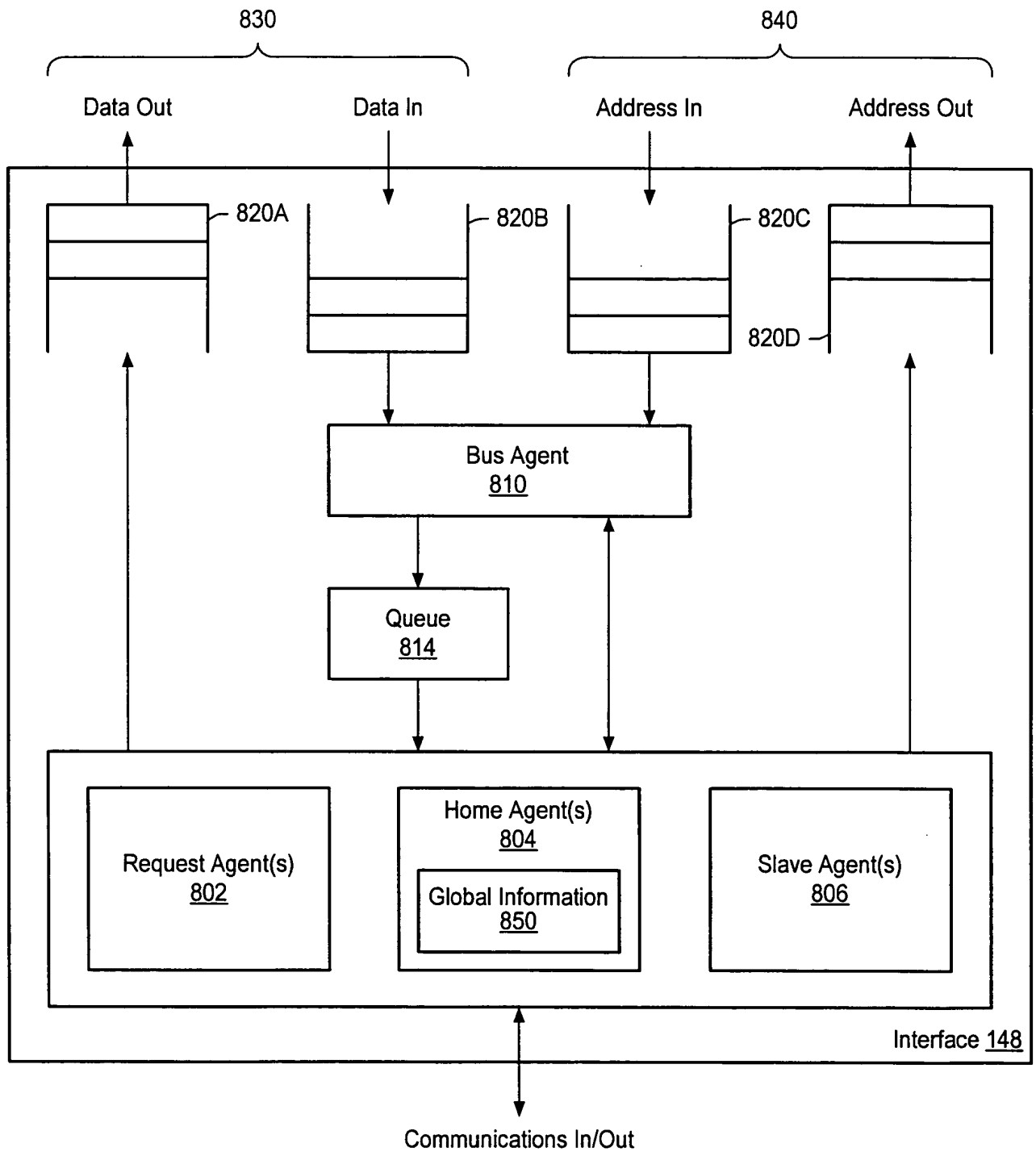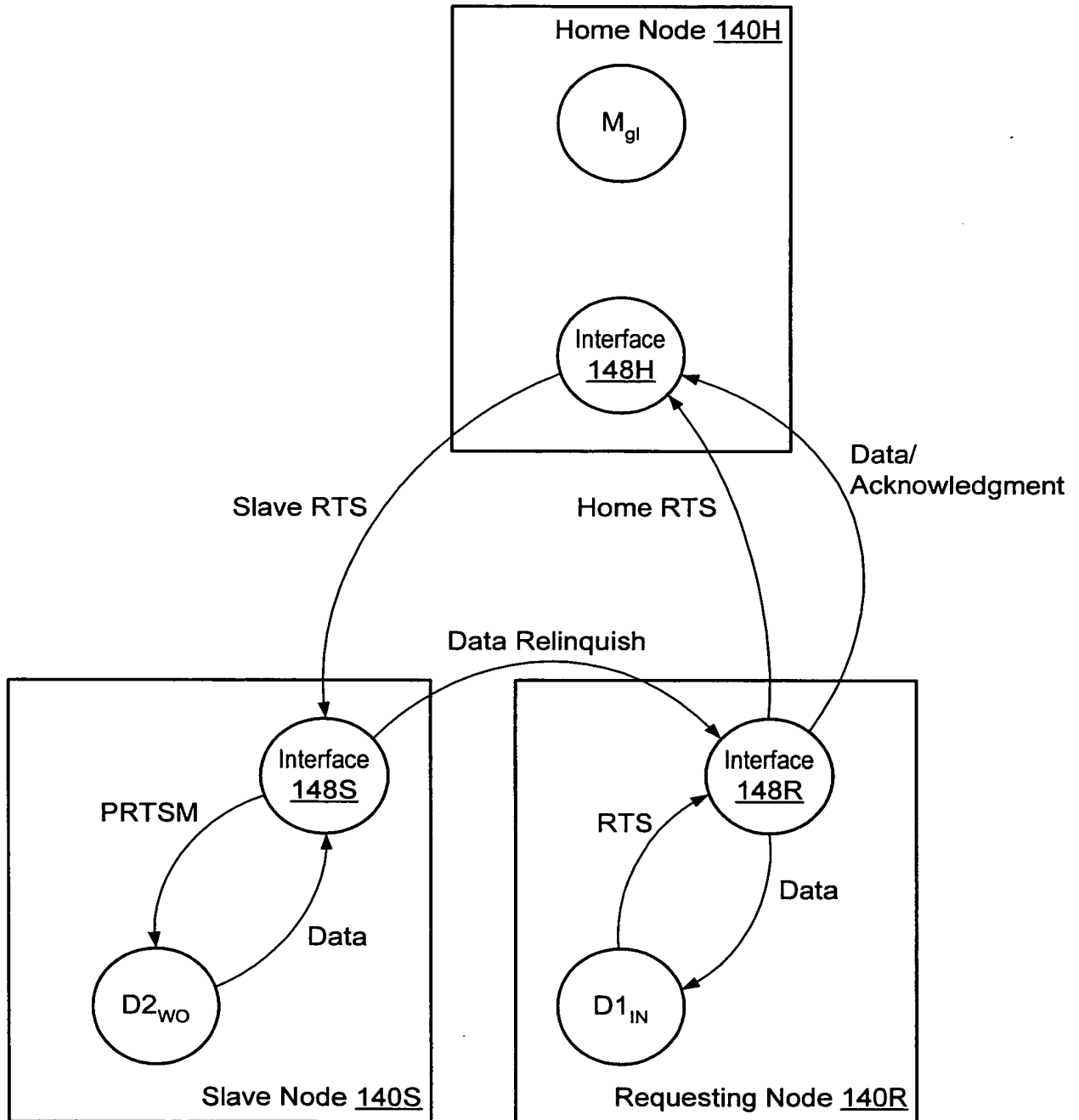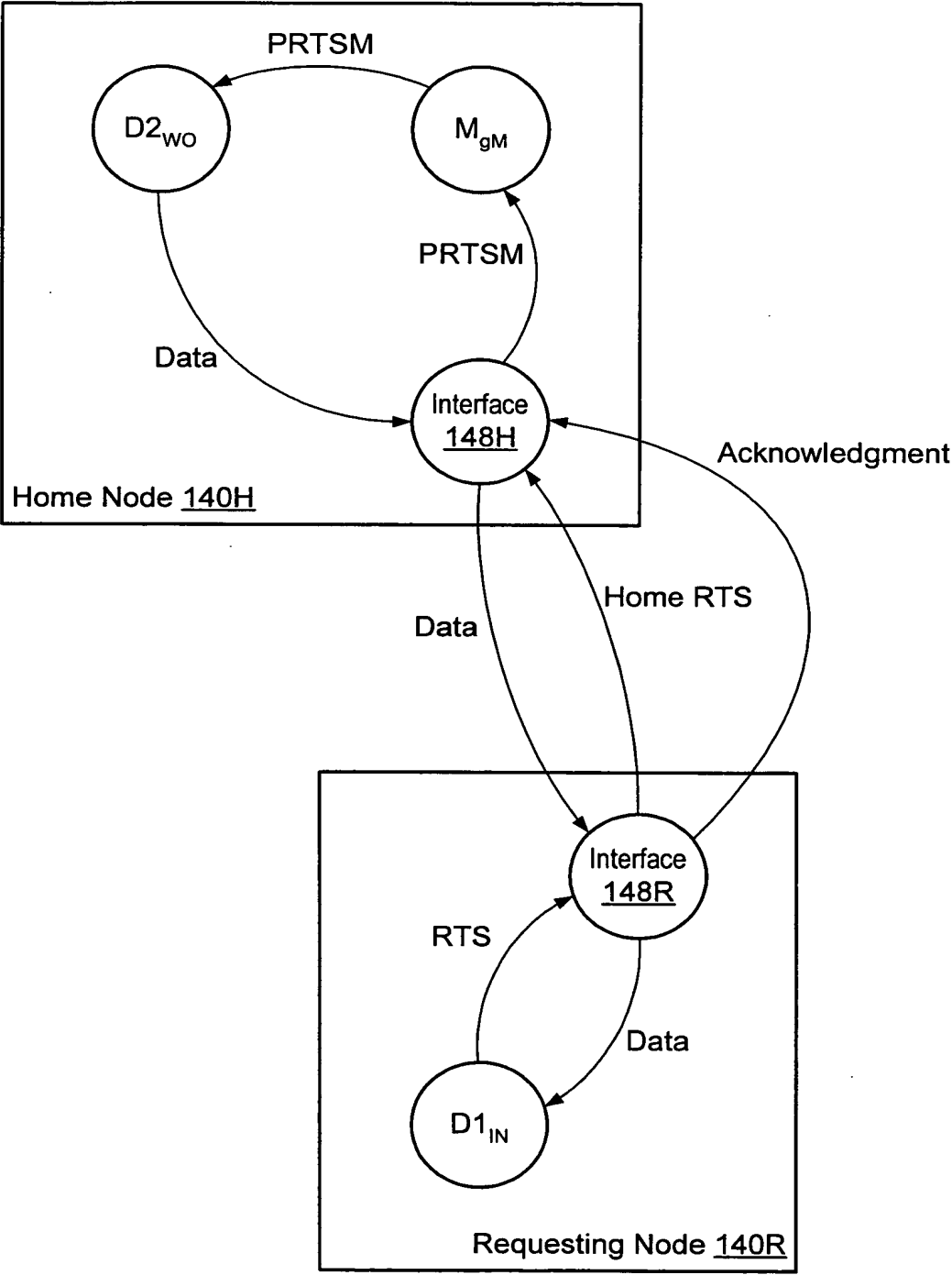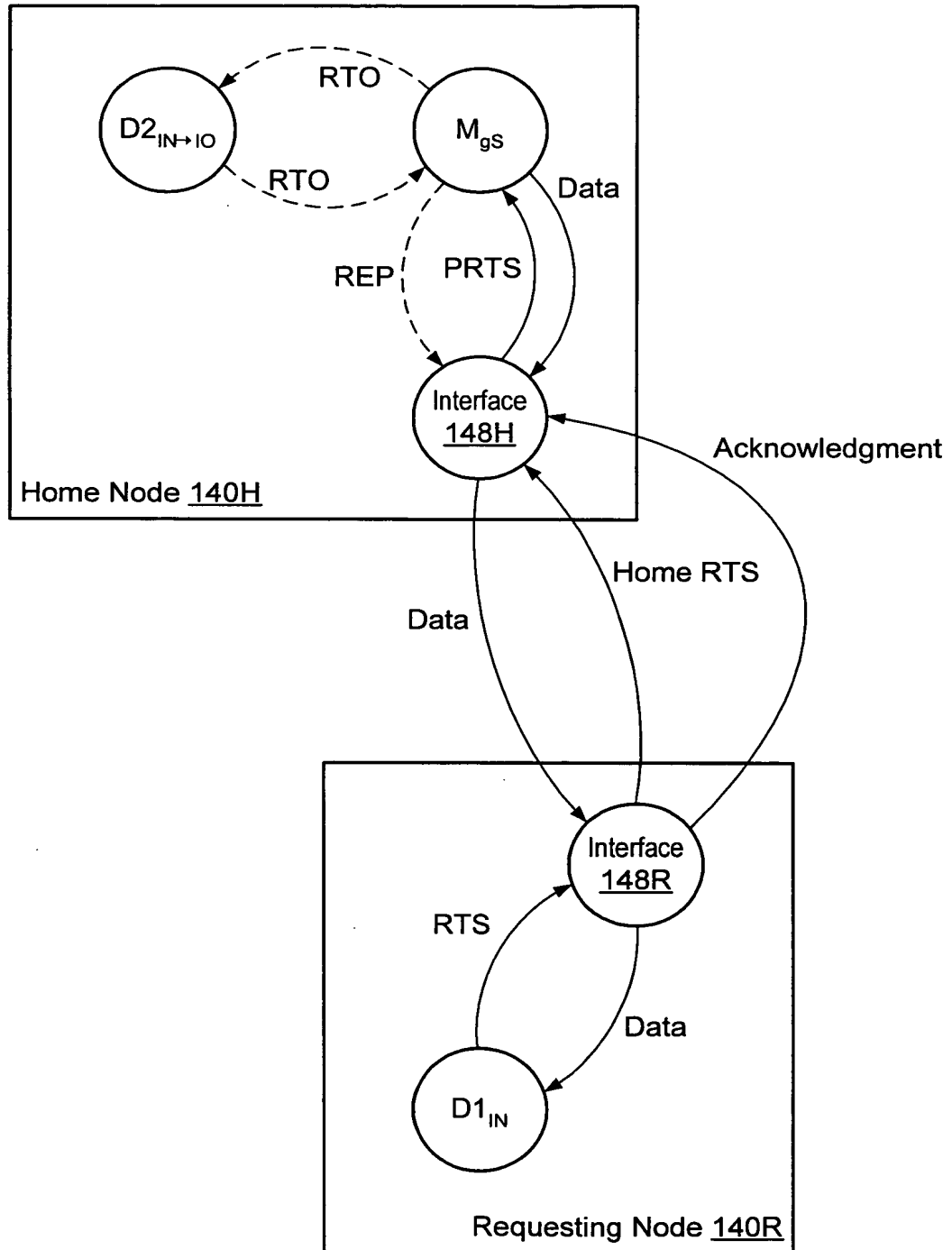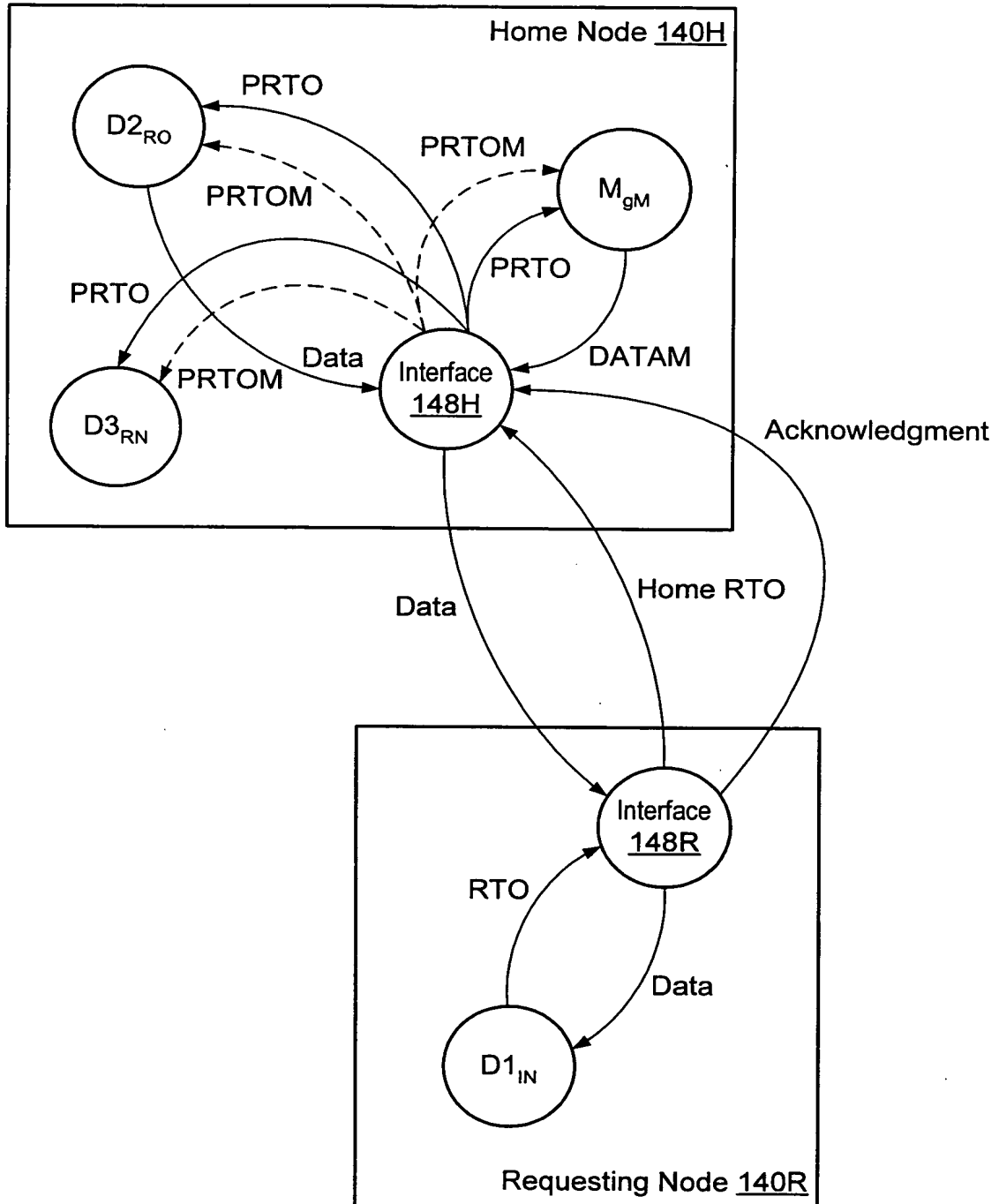
Fig. 33
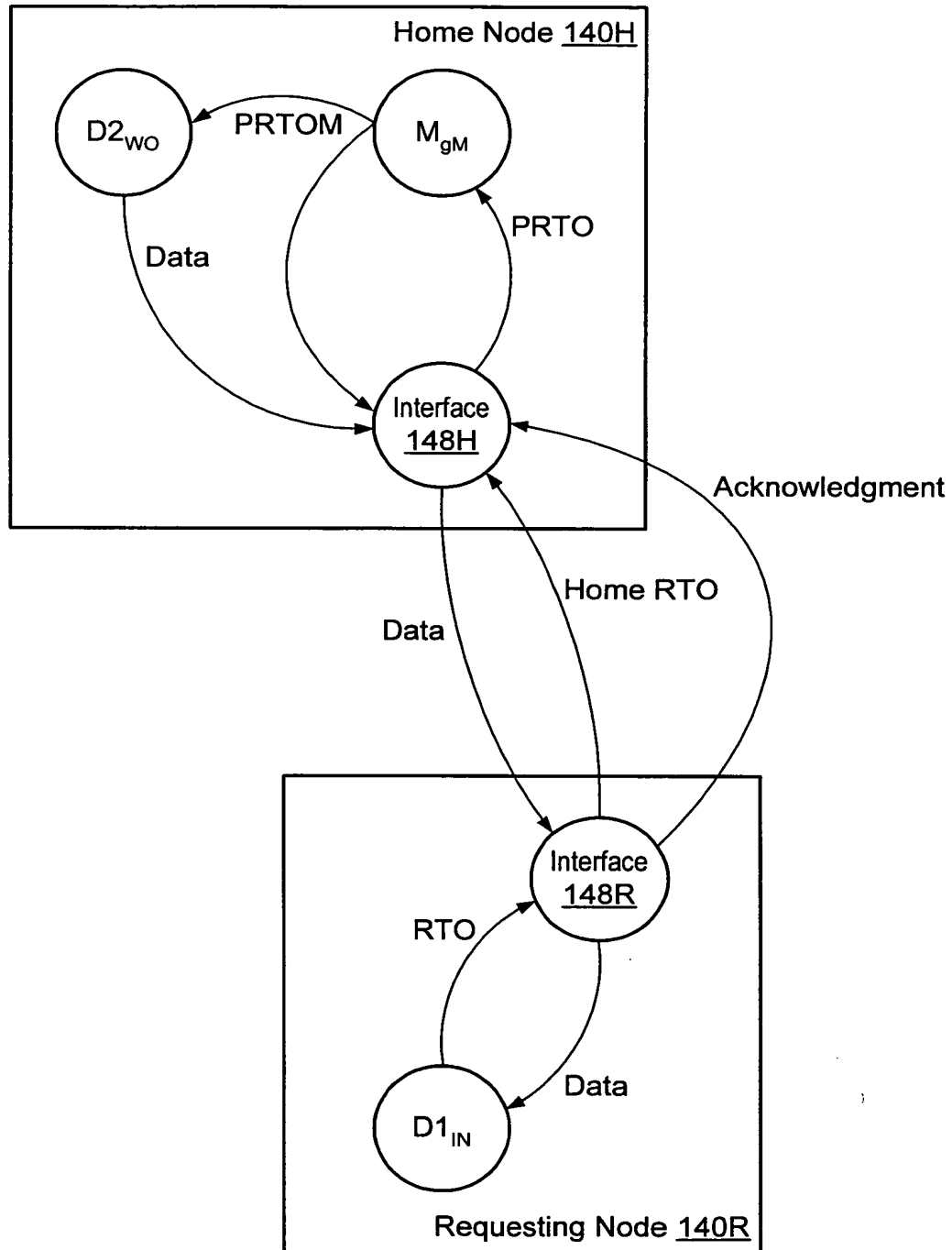
Fig. 34

Response Info	Coherence State	Home Memory Subsystem Action
No	gM	BC Mode-- Allow owning device to respond. PTP Mode-- Forward response to owning device
No	gS	Send REP packet to interface if write access requested
No	gI	Send REP packet to interface
Yes	gM	Respond with copy of the requested coherency unit

Fig. 35

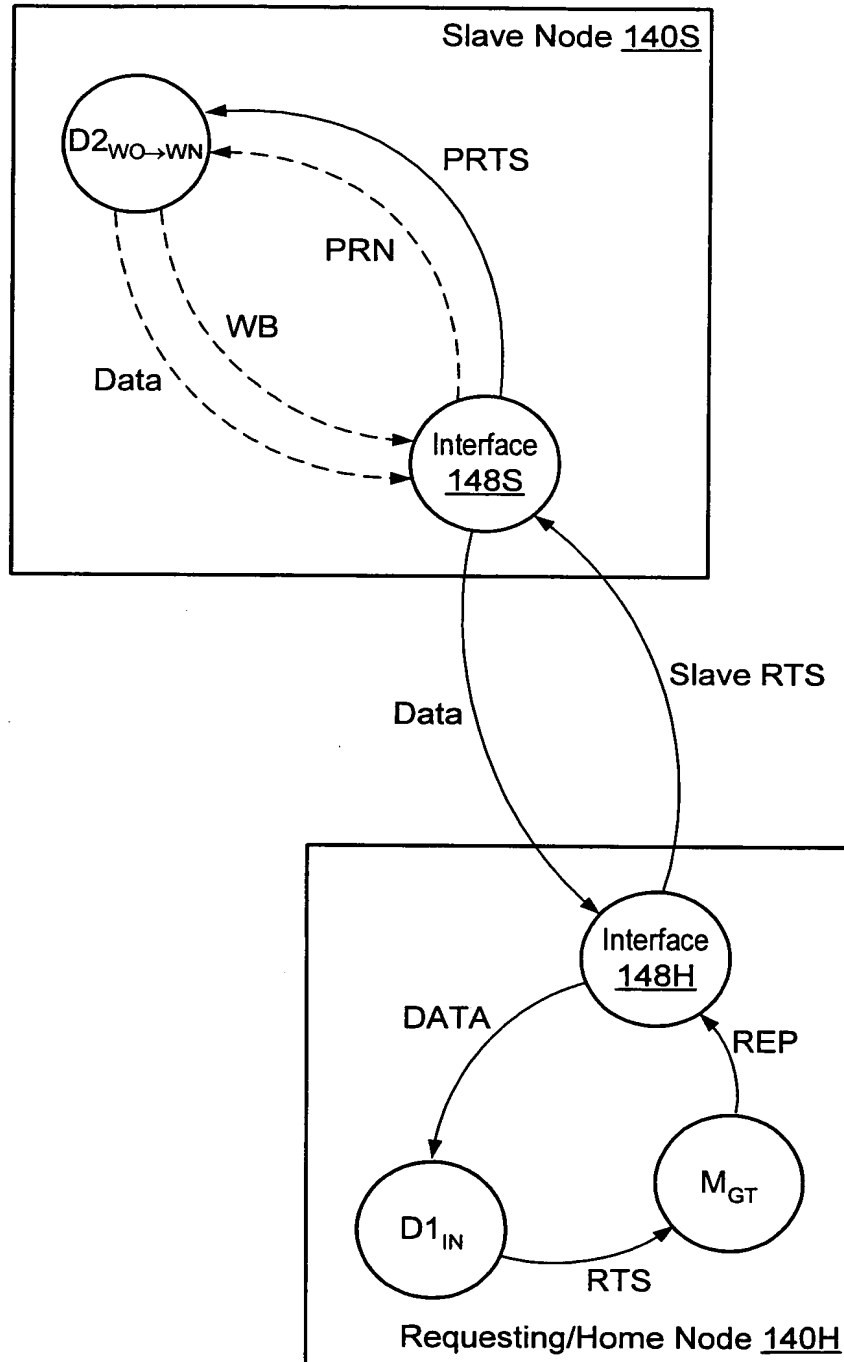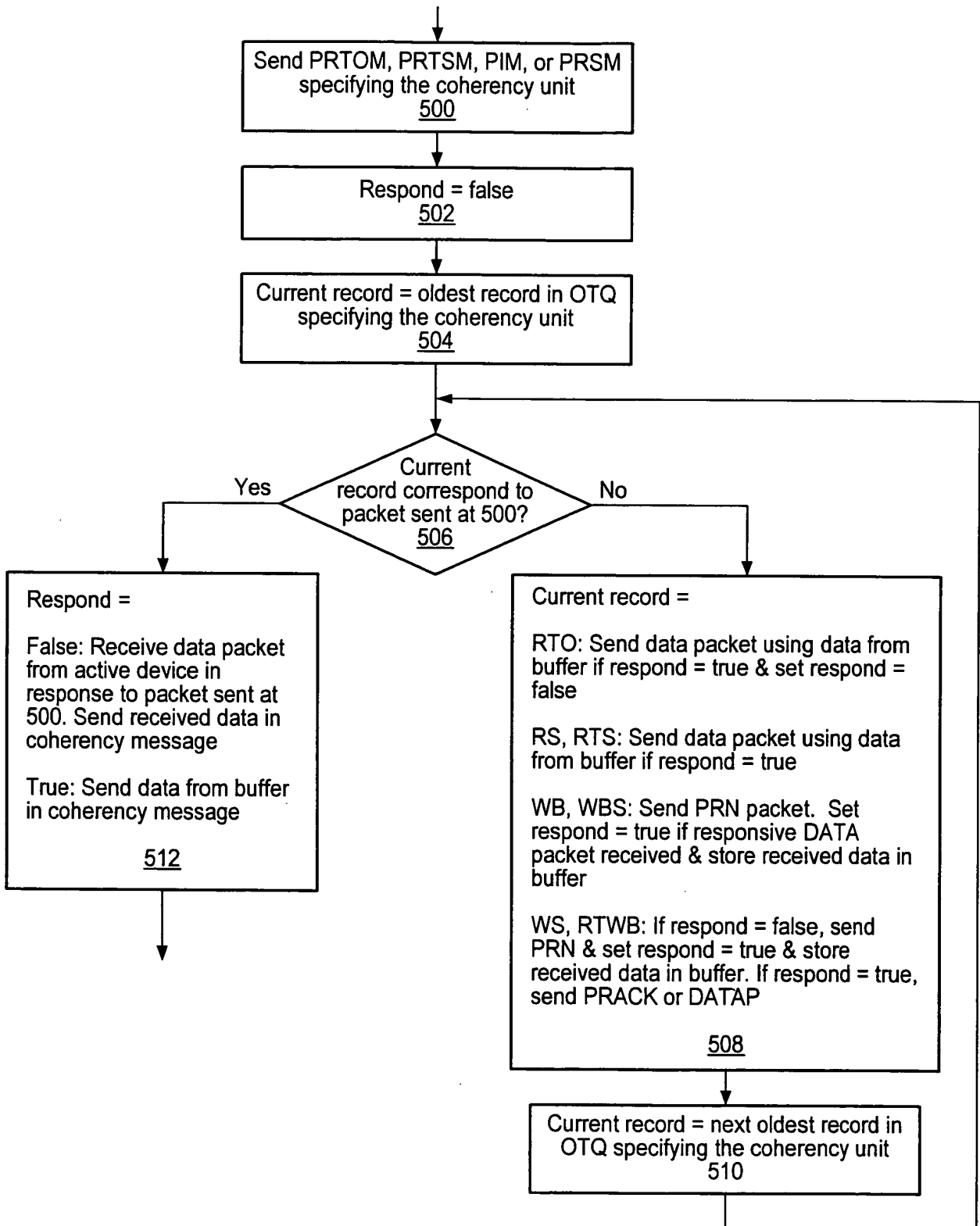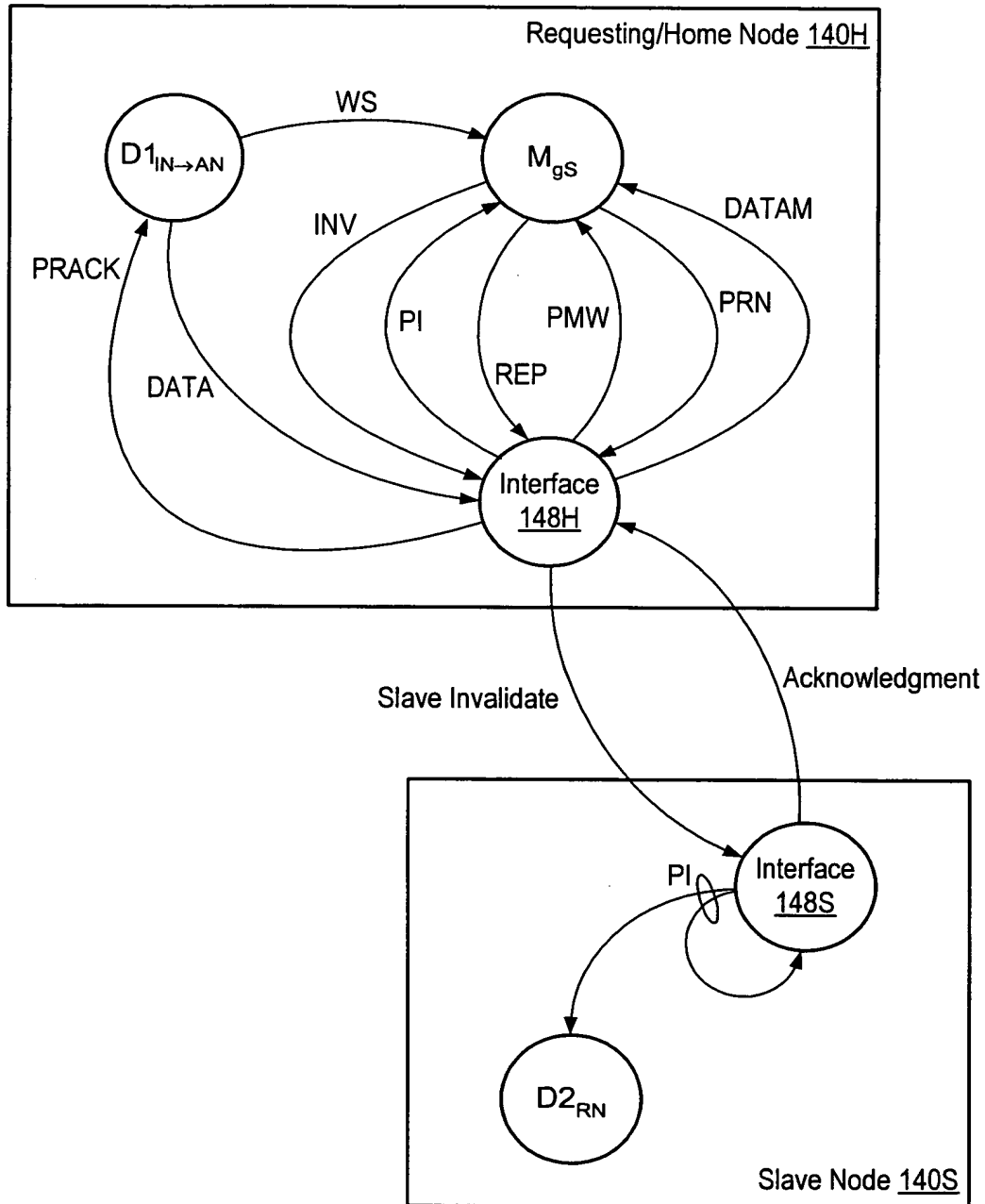| Response Info	Memory Subsystem's Action
mN	Does not respond with copy of coherency unit because a cache within the node owns the coherency unit
mR	Does respond because memory is the owner
mS	Does respond to requests for shared access because memory has shared copy and no active device has ownership; does not respond to requests for write access
mI	Does not respond because memory's copy is invalid

Fig. 36

Fig. 37

Send PRTOM, PRTSM, PIM, or PRSM
specifying the coherency unit
500

Respond = false
502

Current record = oldest record in OTQ
specifying the coherency unit
504

Current
record correspond to
packet sent at 500?
506

Yes

No

Respond =

False: Receive data packet
from active device in
response to packet sent at
500. Send received data in
coherency message

True: Send data from buffer
in coherency message

512

Current record =

RTO: Send data packet using data from
buffer if respond = true & set respond =
false

RS, RTS: Send data packet using data
from buffer if respond = true

WB, WBS: Send PRN packet. Set
respond = true if responsive DATA
packet received & store received data in
buffer

WS, RTWB: If respond = false, send
PRN & set respond = true & store
received data in buffer. If respond = true,
send PRACK or DATAP

508

Current record = next oldest record in
OTQ specifying the coherency unit
510

Fig. 37A

Fig. 38

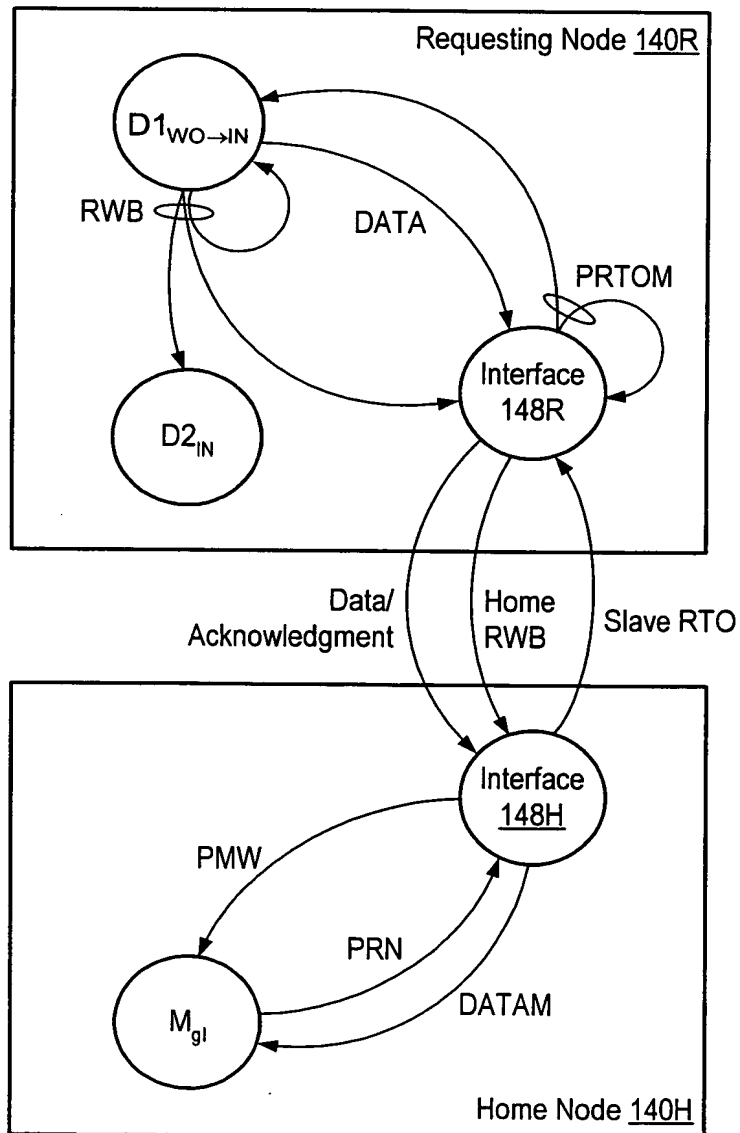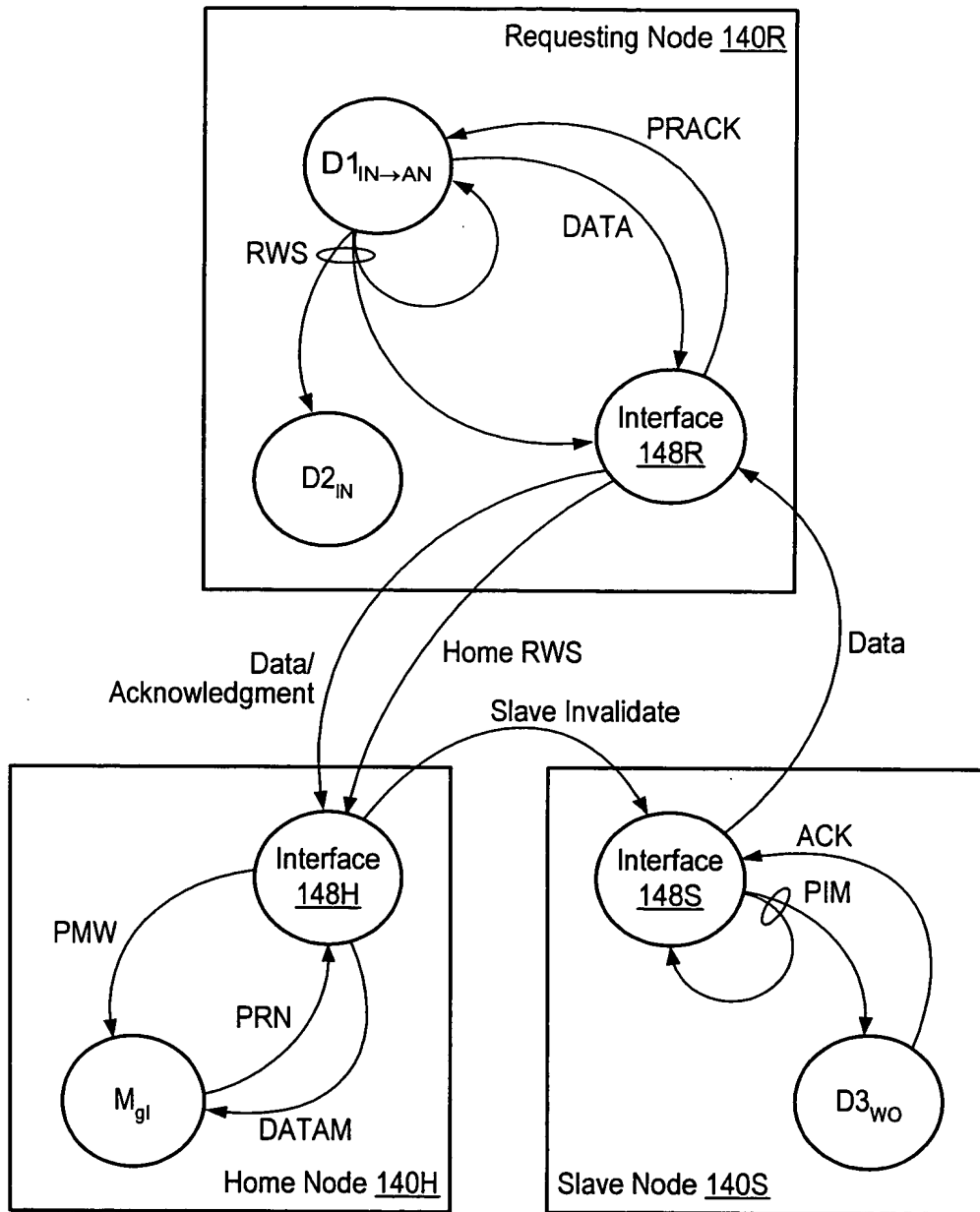| Packet Type	Full Name	Description
RWB	Remote WB	Request sent from an active device in a multi-node system to an interface in order to initiate a WB transaction
RWBS	Remote WBS	Request sent from an active device in a multi-node system to an interface in order to initiate a WBS transaction
RWS	Remote WS	Request sent from an active device in a multi-node system to an interface in order to initiate a WS transaction

Fig. 39

Fig. 40

Fig. 41